The GUTS of HPC's Historic Archive: Grand Unified Text Search (GUTS)

Author: *Wyatt Merians*, Seattle University
Mentor: *Amanda Bonnie*, HPC-DO
LA-UR-19-27107

Abstract LA-UR-19-26912

The High Performance Computing (HPC) division at Los Alamos National Labs has accumulated various HPC documents, records, pictures, and other files over the years and has started storing them into a digital archive. However, the physical location of these materials structured the directory layout of the digital archive to make the physical files easy to locate. Due to this structuring, it is very difficult to find specific topics, information, or files. The goal for GUTS is to be able to search through the files in the archive by text. By integrating a wide variety of text extraction methods to handle the various file types, we will be able to take text from the majority of files in the archive. Then, we will format the extracted text into directory-based databases and use Grand Unified File-Index (GUFI) to provide fast, parallel, search capabilities over the various databases. We will then implement a graphical user interface (GUI) for ease of use. GUTS will ultimately allow the archive to be much more accessible.

HPC's Historic Archive

The Historic Archive has:
- 582 different file types
- 15,900 directories
- 170,041 files

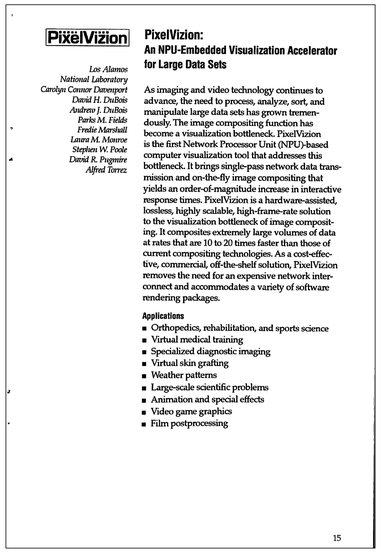
Example of the Directory's Layout After Using GUTS:

```
Book Case 1
    Section 1
        dirGUTS.db
        section 1 file 1.docx
        section 1 file 2.jpg
        section 1 file 3.png
    Section 2
        dirGUTS.db
        section 2 file 1.pdf
        section 2 file 2.pptx
```

Databases & Grand Unified File-Index (GUFI)

There will be a database built in Sqlite3 in each folder, which will allow us to use GUFI.
In every database:
- "Entries" Table
  Filled with:
  - Inode
  - Size
  - Checksum
- "Words" Virtual Table
  Filled with:
  - Id (Basic id by incrementing)
  - Wordf (Words extracted without stop-words, numbers, and punctuation
  - tInode (Same as Inode)
  - tWords (Total number of words)
  - tWordsI (Total number of words without stop-words, numbers, and punctuation)
  - Epochguess (Guess of when the document was created)

Text Extraction



PixelVizion

PixelVizion:
An NPU-Embedded Visualization Accelerator for Large Data Sets

Los Alamos National Laboratory
Carolyn Connor Davenport
David H. DuBois
Andrew J. DuBois
Parks M. Fields
Fredie Marshall
Laura M. Monroe
Stephen W. Poole
David R. Pugmire
Alfred Torrez

As imaging and video technology continues to advance, the need to process, analyze, sort, and manipulate large data sets has grown tremendously. The image compositing function has become a visualization bottleneck. PixelVizion is the first Network Processor Unit (NPU)-based computer visualization tool that addresses this bottleneck. It brings single-pass network data transmission and on-the-fly image compositing that yields an order-of-magnitude increase in interactive response times. PixelVizion is a hardware-assisted, lossless, highly scalable, high-frame-rate solution to the visualization bottleneck of image compositing. It composites extremely large volumes of data at rates that are 10 to 20 times faster than those of current compositing technologies. As a cost-effective, commercial, off-the-shelf solution, PixelVizion removes the need for an expensive network interconnect and accommodates a variety of software rendering packages.

Applications
- Orthopedics, rehabilitation, and sports science
- Virtual medical training
- Specialized diagnostic imaging
- Virtual skin grafting
- Weather patterns
- Large-scale scientific problems
- Animation and special effects
- Video game graphics
- Film postprocessing

15

Photos 1 & 2. Two example files from the Archive

Methodology for a few fie types:

.jpg & .jpeg
- ***Tesseract***
  - Tesseract is an extremely powerful and accurate Optical Character Recognition software. However, it can only be used on a JPG file.

.pdf
- ***pdftotext, PyPDF, PyPDF2, & textract***
  - All extract text from PDF files, but it requires the PDF to have an extractable layer of text or it does not work.
- ***PyOCR & OCRmyPDF***
  - Add the text layer needed by the previous text extraction to PDF files but takes a lot of time, is unreliable, and has to be done manually.
- ***pdf2image, pillow, & Tesseract***
  - The method we are using. We would have to use the other methods for PDF in conjunction and we avoid all their faults by converting to JPG and using Tesseract.

.docx, .pptx, and other Microsoft file types
- ***zipfile & xml.dom.minidom, xml, or another method***
  - This is the next text extraction method I will be working on

Miscellaneous strange file types
- ***Mac Time Machine Backup, a couple installers, corel files, windows dll, audio files, movie files, and many others***
  - These we will most likely have to ignore (though possibly if we have time we can think of a way to integrate them)

Future Work

- Will continue utilizing different text extraction methods for as many of the different file types as possible, starting with Microsoft products like Docs, PowerPoint, Excel, etc.
- Combine all my work on databases and different methods of text extraction in a large wrapper file that runs everything.
- Run GUFI over GUTS and the databases filled with the text from the archive.
- Eventually, create a graphical user interface (GUI) for the entire project.

Potential Graphical User Interface (GUI)

Photo 3. Potential graphical user interface (GUI)

This is a good example of what the GUI for GUTS might look like.
This potential GUI includes:
- The LANL logo that would link to http://int.lanl.gov/
- A possible HPC Historic Archive logo that would link to the homepage
- Publications, Timelines, Videos, and About that would go to their respective links
- A Search Icon that would link to the GUI above
- A search bar that the user would be able to use to search the Historic Archive
- Links that would redirect the user to a more advanced search and help

Acknowledgements

I am grateful for the help of these people:
- Amanda Bonnie
- Garrett Ransom
- Gary Grider

I am grateful for being able to use these softwares:
- GUFI
- Tesseract & PyTesseract
- Leptonica
- Pdf2image
- Pillow
- NLTK (Natural Language Toolkit)
- Sqlite3